REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

Public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington Headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302, and to the Office of Management and Budget, Paperwork Reduction Project (0704-0188), Washington, DC 20503.

1. AGENCY USE ONLY *(Leave blank)*

2. REPORT DATE

October 1994

3. REPORT TYPE AND DATES COVERED

Professional Paper

4. TITLE AND SUBTITLE

SEMANTIC HETEROGENEITY IN DATABASE AND DATA DICTIONARY INTEGRATION FOR COMMAND AND CONTROL SYSTEMS

5. FUNDING NUMBERS

PR: CA60
PE: OMN
WU: IC000083

DTIC
ELECTE
DEC 2 9 1994
S F D

6. AUTHOR(S)

M. G. Ceruti, M. N. Kamel

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)

Naval Command, Control and Ocean Surveillance Center (NCCOSC)
RDT&E Division
San Diego, CA 92152–5001

8. PERFORMING ORGANIZATION REPORT NUMBER

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)

Space and Naval Warfare Systems Command
Washington, DC 20363–5100

10. SPONSORING/MONITORING AGENCY REPORT NUMBER

19941227 100

11. SUPPLEMENTARY NOTES

12a. DISTRIBUTION/AVAILABILITY STATEMENT

Approved for public release; distribution is unlimited.

12b. DISTRIBUTION CODE

13. ABSTRACT *(Maximum 200 words)*

Semantic heterogeneity has been investigated in connection with the database and data dictionary integration efforts that support Command, Control, Communications and Intelligence (C^3I) systems. Based on this investigation, a systematic approach to the resolution of semantic heterogeneity has been developed and illustrated with examples derived from the component C^3I system databases in a federation. A methodology is introduced for resolving semantic conflicts to construct a tightly coupled federated database system by facilitating the development of a global schema derived from the individual schemas of the component databases. This methodology of resolving semantic conflicts results in the formation and modification of synonym-homonym groups (SHG), a concept introduced and developed in the paper. A detailed analysis using a three-phased procedure is introduced, with each phase exploring semantic heterogeneity at progressively finer levels of information granularity. For the purpose of illustration, the simplest case of a two-component database integration into a tightly coupled federated database system is considered, but the methodology can be generalized to include three or more component databases in a federated system. It also can be applied in the case of a fully merged database integration involving any number of component databases. A variety of inconsistencies were identified using the heuristics implemented in the algorithm. Resolutions of the problems arising from semantic heterogeneity are suggested, and directions for future research are explored.

DTIC QUALITY INSPECTED 8

~~Published in~~ *Proceedings of the Eleventh DoD Database* ~~*Colloquium*, August 1994~~

14. SUBJECT TERMS

Command and Control
Afloat Systems
Correlation
Intelligence

15. NUMBER OF PAGES

16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT	18. SECURITY CLASSIFICATION OF THIS PAGE	19. SECURITY CLASSIFICATION OF ABSTRACT	20. LIMITATION OF ABSTRACT
UNCLASSIFIED	UNCLASSIFIED	UNCLASSIFIED	SAME AS REPORT

Semantic Heterogeneity in Database and Data Dictionary Integration for Command and Control Systems

Dr. Marion G. Ceruti
Naval Command, Control and Ocean Surveillance Center, RDT&E Divsion
San Diego CA, 92152

and

Dr. Magdi N. Kamel
Naval Postgraduate School
Monterey CA, 93943

esion For
CRA&I ☑
TAB □
nced □
ation
A-1

ABSTRACT

Semantic heterogeneity has been investigated in connection with the database and data dictionary integration efforts that support Command, Control, Communications and Intelligence (C^3I) systems. Based on this investigation, a systematic approach to the resolution of semantic heterogeneity has been developed and illustrated with examples derived from the component C^3I system databases in a federation. A methodology is introduced for resolving semantic conflicts to construct a tightly coupled federated database system by facilitating the development of a global schema derived from the individual schemas of the component databases. This methodology of resolving semantic conflicts results in the formation and modification of synonym-homonym groups (SHG), a concept introduced and developed in the paper. A detailed analysis using a three-phased procedure is introduced, with each phase exploring semantic heterogeneity at progressively finer levels of information granularity. For the purpose of illustration, the simplest case of a two-component database integration into a tightly coupled federated database system is considered, but the methodology can be generalized to include three or more component databases in a federated system. It also can be applied in the case of a fully merged database integration involving any number of component databases. A variety of inconsistencies were identified using the heuristics implemented in the algorithm. Resolutions of the problems arising from semantic heterogeneity are suggested, and directions for future research are explored.

Proceedings of the Eleventh Annual DOD Database Colloquium,
"Database Integration and Innovation", sponsored by AFCEA, 29 - 31 August, 1994

I. INTRODUCTION

The field of interoperability and integration of heterogeneous databases, including the area of multidatabase and federated database systems, is among the most active areas of database systems research [1-8]. Interoperation and integration between databases, the data dictionaries that describe them, and the application systems that access them depend on resolving heterogeneity or incompatibilities at different levels. These levels include at least the following three: 1) Platform heterogeneity which includes incompatibilities of hardware, operating systems, transaction management, networking protocols, etc., 2) Data structures, languages, and constraints heterogeneity of the different information systems that manage user applications, and 3) Semantic heterogeneity which is likely to be present, since application systems were designed independently by different people who have different perspective on the real world they are trying to model.

In this paper we address the third level, that of semantic conflicts identification and resolution. Sheth and Larson defined semantic heterogeneity as the existence of disagreement about the meaning, interpretation, or intended use of the same or related data [1], a definition also assumed throughout this paper. Here, we continue the heterogeneity classification process to include three distinct, but related sublevels within the category of semantic heterogeneity.

Semantic heterogeneity issues are becoming increasingly important in the databases of military command and control systems because of the trend toward integrating systems with similar or complementary functions. These integration efforts are undertaken to eliminate duplication of effort and to consolidate and enhance current capabilities. In an environment characterized by constant change, developers are frequently faced with the task of merging systems that originally were developed separately, by different organizations for dissimilar purposes, using different software tools to manipulate what logically should be the same or similar data, but from different origins. To complicate the problem further, the users of command and control systems demand a seamless homogeneous, current, consistent, and accurate picture of the real world as represented in the supporting databases of their command and control systems. The fact that the logical database actually consists of a collection of previously autonomous databases should be transparent to the users and to the application [2]. Whereas some problems of semantic heterogeneity are very difficult to solve, true interoperation can be achieved only when all significant semantic heterogeneity issues are addressed [3].

The scope of the present work is limited to review of literature on semantic heterogeneity; in-house studies concerning relational databases at Naval Command Control and Ocean Surveillance Center (NCCOSC) Research

Development Test and Evaluation Division (RDTE DIV) and at the Naval Postgraduate School (NPGS); case studies of the integration of previously autonomous command, control, communications, computers and intelligence (C^4I) systems. Proposed solutions to problems of data dissimilarities and mismatches are discussed vis-a-vis actual military database integration efforts. This paper cites examples of semantic heterogeneity derived from databases of Department of Defense (DOD) tactical systems; however most, if not all, of the concepts described herein can apply to non-tactical systems as well. A three-phased methodology is presented for identifying and resolving semantic heterogeneity. The algorithms of this methodology are depicted in a series of trouble-shooting flowcharts.

The organization of the paper is as follows. Section II discusses different approaches to database integration and the role of the data dictionary in the integration. Section III discusses the levels of heterogeneity in databases with emphasis on semantic heterogeneity. Section IV presents examples of semantic heterogeneity from previous C^4I database integration efforts. Section V develops a comprehensive approach for identifying and resolving semantic heterogeneity during integration efforts. In this section, the three sublevels within the category of semantic heterogeneity are explored during the conflict resolution process. Finally, Section VI concludes our paper with a summary and directions for future research.

II. APPROACHES TO DATABASE INTEGRATION AND THE ROLE OF THE DATA DICTIONARY

In this section we discuss several approaches for database integration and the role of the data dictionary in the integration. A data dictionary is a collection of facts about objects or events in the database environment [9]. In relational systems, the data dictionary consists of one or more relations containing at least the following data-related attributes: relation name, attribute name, data type, data length, data definition, and the units of the data. The data dictionary also contains comprehensive information about database structure. In addition, the more useful data dictionaries provide information to indicate whether an attribute is part of the primary key, whether or not it has an index , and whether or not null values are allowed.

Sometimes data dictionaries list other information that can be used to sort the data into categories of originating authority, use, subject matter, access control, releasability, and system partition. In addition to complete metadata, other desirable features can be found, such as an alphabetical cross listing of attributes with relations, as well as an alphabetical listing of relations with attributes. Some databases include certain metadata in relations for security and administrative-tracking purposes, as is the case with the Naval Warfare Tactical Database and the Operations Support System (OSS) Integrated Database (IDB) [10].

During the integration of two or more databases, semantic heterogeneity can occur with respect to any relation or attribute in the data or metadata relations. Dealing with consequences of semantic heterogeneity can range from the trivial and minor to the serious and significant.

To increase data access and sharing among different databases, we can identify at least three approaches: full merging, the tightly coupled federated database systems (FDBS), and the loosely coupled FDBS. Both approaches to the formation of FDBS are subsets of the multidatabase approach described in [1]. Litwin et al. also have used the term "multidatabase system" as synonymous with "loosely coupled FDBS" [11], which serves to illustrate the state of confusion concerning the terminology in the literature of this field. The first approach is that of full merging of the databases of interest. In this approach, databases are integrated by combining relations into a single, physically unified database. In this type of integration, the data dictionary is integrated in a manner similar to that of the database itself. Such an approach was used in the integration of some C^4I databases [10, 12].

Frequently, however, it is desirable to provide data integration without sacrificing the autonomy of individual databases [1]. This has led to the federated, multidatabase approaches. Under the tightly coupled, federated database approach, each independent database is considered a logical component in the federation. These components are connected by one or more global schemas that represent the integration of several local schemas. The global schemas, therefore, represent information that can be shared by the federation components. This concept is also expressed by the term "federated schema" [1]. This tightly coupled, federated approach maintains the autonomy of individual databases but constructs a global schema that hides the distribution and heterogeneity of the underlying databases. A user can issue queries on the global schema to retrieve information that resides on several physical component databases as if he or she is accessing a single database. The specification of the global schema is maintained by a global controller that acts as a coordinator among the database components of the federation as well as a translator. It receives a query on the global schema, decomposes it, and translates into subqueries on the individual schemas for processing. When processing is complete, the controller collects the results, identifies and resolves data conflicts, reformats and sends the result back to the requesting user. It is important to note that a global schema is a virtual one since it does not correspond to a physical database.

Under the loosely coupled approach, a global schema is not constructed. Rather, users are aware of the existence of distinct databases, but can access them using a common language or interface. These languages or interfaces allow the joining of data in different databases, the broadcasting of user queries over a number of databases, the exchange of data between databases, and the dynamic transformation of attribute values, unit of measures, etc.

A data dictionary in a federated or multidatabase system differs from that in homogeneous systems managed with a single database management system (DBMS). For single-database systems, the data dictionary describes not only the logical data structure with its metadata, but also the underlying database. Whereas the data dictionary in a federated or multidatabase system is still expected to present a global picture of the data, it may not describe the individual database structure defined and maintained in the local data dictionaries.

There is an increasing trend toward both the loosely coupled and the tightly federated, multidatabase systems approach for integrating databases with considerable platform, operating system and DBMS heterogeneity. In this case, database integration is treated as a process more distinctly removed from the database and data dictionary integration in the full merging approach. Even though data from the different systems remain autonomous and distributed on different platforms with different DBMS, the metadata, however, need to be integrated into a single virtual global data dictionary describing the federated or multidatabase database view of the system.

III. LEVELS OF HETEROGENEITY AND SEMANTIC HETEROGENEITY IN DATABASES

As discussed in section I, three levels of heterogeneity need to be addressed whenever data from different sources are integrated, either in multidatabase, federated system, or within a single, homogeneous database. These are platform, data model, and semantic levels. For example, data model heterogeneity arises from differences in DBMS that manage different databases, whereas homonyms and synonyms are an example of heterogeneity at the semantic level. The following are some examples of heterogeneity at each main level of our classification [2-4, 7, 8]:

1. Platform Heterogeneity
   a. DBMS vendors (Sybase vs. Oracle, for example)
   b. DBMS transaction processing algorithms (locking, time stamping, validation, concurrence control)
   c. DBMS query processing (processing and optimizing strategies)

2. Data Model Heterogeneity
   a. Schemas or data models
   b. DBMS query languages and versions

c. Integrity constraints (discretionary vs. mandatory security constraints)
d. Nullness requirements and other attribute constraints

3. Semantic Heterogeneity
a. Conceptual schema (metadata specification)
b. Data security classification levels (U, C, S, vs. U, S)
c. Relation and attribute names and definitions (application-specific terminology, homonyms, synonyms)
d. Ranges and domains of data elements (database content)
e. Data element format, type (CHARACTER, NUMBER, etc.) and length
f. Units of measure (nautical miles vs. kilometers)
g. Levels of precision - (3.5 meters vs. 3.54 meters)
h. Levels of granularity (squadron, unit, fleet)
i. Data inconsistencies (different data element values reported for the same attribute in the same table in different database systems.)

Within the category of semantic heterogeneity, some conflicts can occur at the conceptual, schema level, such as synonyms and homonyms, whereas data-level conflicts arise from differences in the data values returned by different databases for the same objects. A detailed classification of semantic conflicts with examples from Naval Administrative Databases is presented in [7].

In the present work, the discussion is limited to semantic conflicts occurring at the schema level as well as data level conflicts that can be determined at schema-definition time. These include homonyms and synonyms, differences in data types, length, units of measure, and levels of precision, as well as differences in data ranges and domains.

Figure 1 is a graphic representation of the various levels of heterogeneity, including semantic heterogeneity, which is divided into two levels: schema and data level heterogeneity. Schema-level heterogeneity is itself divided into three sublevels, each with progressively finer granularity. These levels were chosen as a logical progression to simplify and facilitate the development of the algorithm described in the Section IV, which is a blueprint for the systematic identification and resolution of semantic schema-level conflicts.

Sublevel one, the relation level, contains database components at the coarsest level of granularity. This sublevel is limited to semantic heterogeneity involving the names and definitions of relations, both in comparison to the names and definitions of other relations, as well as in comparison to those of attributes. The resolution of semantic inconsistencies at sublevel one does not require access to the data fill.

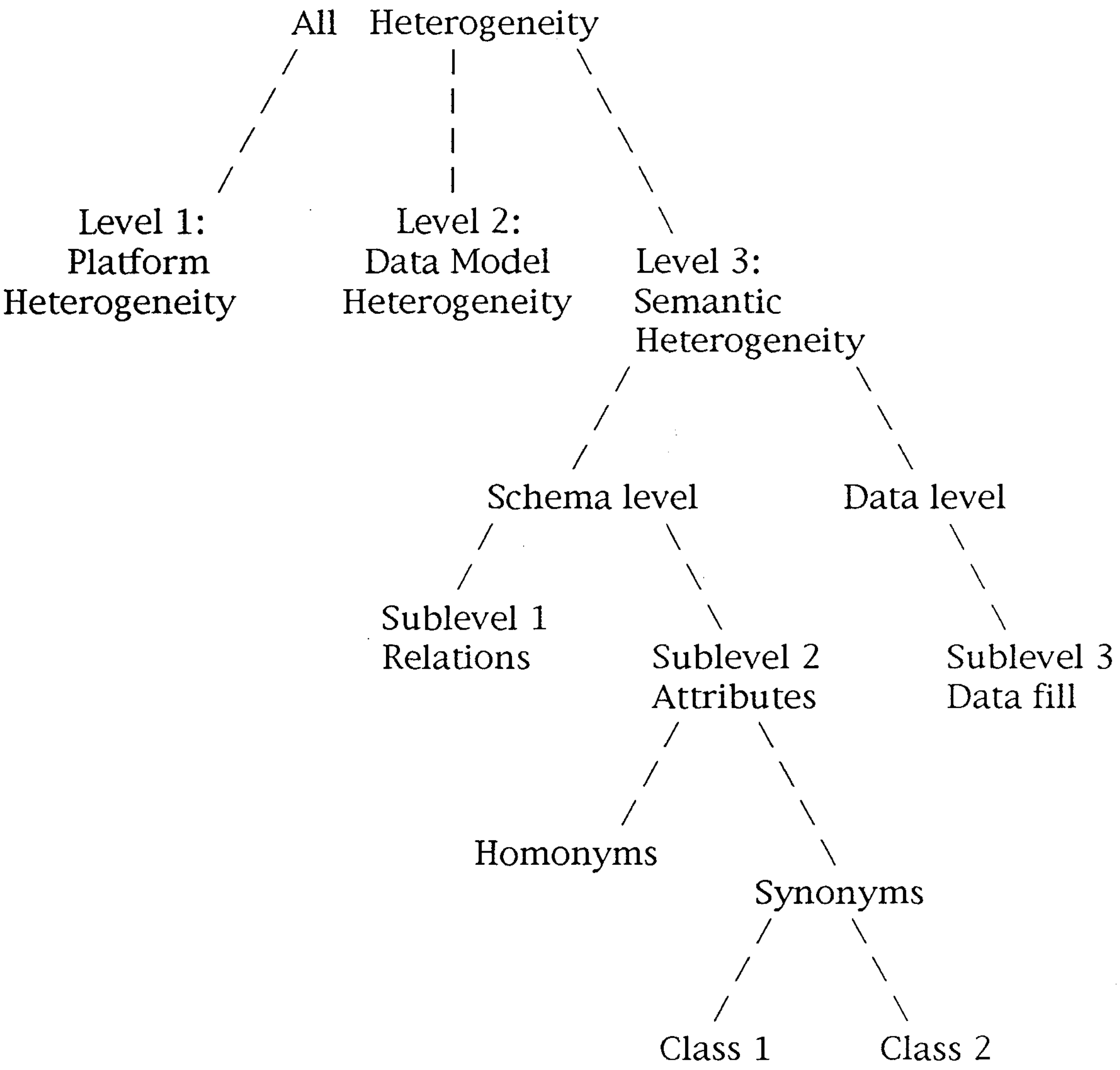

Figure 1. Categories of heterogeneity encountered during database integration.

Sublevel two, the attribute level, is characterized primarily by heterogeneity arising from the properties of attributes, such as data element names, definitions, meanings, data types and lengths. For example, a homonym problem occurs when different real world objects (e.g. entities and attributes) have the same name in different databases. A synonym problem occurs when the same real world entity is named differently in different databases. In general, the risk of homonyms is higher when the vocabulary of terms is small, whereas the risk of synonyms is higher when the vocabulary of terms is rich. Also, the risk of synonyms increases, if two users adopt vocabularies at different abstraction level [13]. Most analysis at sublevel two can be performed using queries on the metadata, without consulting the data fill.

Data-type conflicts occur when equivalent real world attributes have different data types (e .g., character vs. numeric). Similarly, length conflicts occur when equivalent real world attributes have different lengths. Type conflicts are quite common when dealing with databases designed for different implementations, whereas length and range conflicts are more likely to occur as a result of semantic choices [7].

In an on-line data dictionary derived from the integration of one or more databases, all instances of semantic heterogeneity at sublevel one and most at sublevel two can be discovered by analyzing the results of appropriate queries on the metadata. For example, a cross listing of metadata by attribute names in alphabetical order can highlight homonyms. Detecting synonyms, however, is not quite as simple as detecting homonyms, because the wording of data definitions can vary while the meanings remain identical. That not withstanding, if definitions are worded identically, synonyms can be detected easily. A detailed working knowledge of the data also is helpful in identifying synonyms.

Finally, sublevel three which concerns the data fill, the level of the finest granularity, is necessary because many semantic conflicts cannot be resolved at the schema level. All detection of semantic heterogeneity at sublevel three, the data fill level, require access to the data fill to compare data element ranges, domains, units of measure, levels of precision, and data element values. For example, to identify semantic heterogeneity with respect to data range and domain, queries on the data fill must be performed. Range and domain conflicts occur when equivalent real world attributes have different allowable range definitions in different databases. Domain heterogeneity refers to differences between databases because of different ranges and domains of attributes that are supposed to represent the same or closely related entities. The fill can differ independent of any other kind of heterogeneity. In the next section, we give some examples of semantic heterogeneity derived from previous and current C^4I integration efforts.

Moreover, different levels of abstraction exist, depending on how far removed the data representation is from the real-world entity. This concept of an entity as expressed by a database "proxy" that can have multiple representations also has been described by Kent [6]. Kent also has identified some issues that arise in multidatabases from entity identity and naming [6].

We introduce the concept of two categories to express synonym abstraction. A class-one synonym occurs when different attribute names represent the same, unique real-world object or concept using the same data type, length, range and domain. In contrast, a class-two synonym occurs when different attribute names, as expressed with different data types, lengths, ranges, and domains point uniquely to the same real-world entity. These classes are also shown in Figure 1.

If attributes have the same domain, they become potential class-two synonyms. Class-two synonyms cannot be resolved at the schema level, and in many cases at the data-fill level, even though potential class-two synonyms can be identified at sublevel two (Fig. 1). Another level is needed that accounts for how data are updated and implemented, which is outside the scope of the present work. Class-one synonyms, however, can be resolved using the algorithms presented in Section V.

A distinction can be made between the definition of a relation or attribute, and its meaning. The definition refers to the exact wording found in the data dictionary to describe a relation or attribute, whereas the meaning refers to the interpretation of definitions. An example of two attributes that have different definitions but identical meanings is shown in Table 4, in which SECUR is defined as "Security code" in the SORTS_SPCAP relation, and as "Security classification" in the SORTS_TSKCD relation. These definitions differ whereas the meanings are the same. Synonyms in both classes one and two frequently will have different definitions with the same underlying meaning.

It should be noted that the levels and sublevels of heterogeneity are interdependent. For example, platform heterogeneity can lead to semantic heterogeneity. To illustrate, given a DBMS A that supports data types X, Y, and Z whereas DBMS B supports only data types X and Y, how to handle data type Z when integrating these two databases becomes an issue of resolving a schema semantic conflict. Hence, a DBMS could contribute to or solve problems of semantic heterogeneity by the number of allowed data types. If the DBMSs that manage the databases to be integrated provide many specific data types such as date, latitude, longitude, some problems of semantic heterogeneity would be resolved because the added specificity imposed by some data types can remove the ambiguity and facilitate the integration. An example of how semantic heterogeneity can arise from DBMS heterogeneity is presented in Section IV.

IV. SEMANTIC HETEROGENEITY CASE STUDIES

Database developers and data engineers of command and control systems are concerned with the semantic implications of the integration of databases and their corresponding metadata. As indicated in the previous section, homonyms, synonyms, as well as inconsistencies in data type and length are among the types of semantic heterogeneity that can occur. In this section, some real-world examples from operational and developmental C^4I systems' databases are presented.

Databases of Command and Control Systems

The Fleet Command Center Battle Management Program (FCCBMP) was originally designed to support expert systems for the Commander in Chief, Pacific Forces (CINCPACFLT). The FCCBMP Integrated Database (IDB) used data related to force requirements and capabilities from a variety of sources, including the Operations Support Group Prototype, and the Technical Database (TDB). The FCCBMP IDB was developed with the Oracle DBMS [12].

The Operations Support System (OSS) is a command and control decision -aid program sponsored by the Space and Naval Warfare Systems Command. The purpose of OSS is to provide command centers with a hybrid information management C^4I system developed using evolutionary acquisition methodology. The system was designed to evolve in stages by capturing the functionality from a variety of other C^4I systems, including the FCCBMP, and integrating these functions into a single, cohesive unit with a uniform look and feel. The structure of the OSS IDB was patterned after databases developed for several other Department-of-Defense programs [10]. The OSS IDB also was developed with Oracle DBMS.

Part of OSS IDB came from the FCCBMP IDB, and part was developed in accordance with standards from the Naval Warfare Tactical Database (NWTDB), the standard, authoritative information source for all Naval warfare systems [10]. The NWTDB includes the Naval Intelligence Database (NID); the Military Intelligence Integrated Data System/Integrated Database (MIIDS/IDB) unit, location and facility data sets; the Navy Tactical Command System Afloat (NTCS-A) air-tasking order data sets; and the OSS track, readiness, operational-area, and fixed-site data sets. The NID is maintained using Oracle, whereas Sybase is the DBMS of choice for NTCS-A.

Some aspects of the database integration now in progress for NWTDB can serve as a model for the database integration of the Joint Maritime Command Information System (JMCIS), which will include many data sets from NWTDB, as well as the databases required to support a wide variety of maritime C^3I applications with diverse DBMSs from the U. S. Navy, Marine Corps, and Coast Guard. The OSS IDB and the NTCS-A database will be integrated into the JMCIS Federated Database (FDB). The MIIDS/IDB as well as the NID component of the NWTDB will be included. In the present work, these database integration efforts provided metadata for case studies in integrating data dictionaries and identifying semantic conflicts.

Examples of semantic heterogeneity in the NWTDB are presented below. The component databases of NWTDB were chosen for inclusion in the standard because they are relatively advanced and well developed, they cover tactical subject matter, and they have widespread use and visibility throughout the Navy and DOD. NWTDB was chosen for analysis in these case studies not only because of its importance to the Navy and Marine Corps, but also because of its superior organization and documentation. Many examples of semantic heterogeneity were observed readily in NWTDB that would otherwise have been time consuming and difficult to identify if the component databases had been chosen and organized in a less systematic manner. Thus, it is not the intention of the authors to single out the NWTDB for criticism, because the semantic inconsistencies discussed herein can be expected to occur in any database integration of this magnitude.

Data Analysis

These metadata are summarized in Tables 1 through 4. The examples in Tables 1 and 2 came from the FCCBMP IDB, whereas Tables 3 and 4 were derived from NWTDB tables, including the Naval Intelligence Database (NID), Operations Support System (OSS) Integrated Database (IDB), NTCS-A, and the MIIDS/IDB, as included in the NWTDB. Table 4 lists the component databases from which the attributes in Table 3 were derived. Because the OSS IDB itself results from an integration of several different data sources, each general OSS data category is represented explicitly in the NWTDB, and also in Table 4.

The attribute definitions in Table 1 can be confusing when taken out of context [12]. For example, in the relation, EIC_NAME_MISSION, which correlates equipment identification codes with the warfare mission areas in which the equipment is used, the domain of attribute AAW consists of "yes" and "no". The fill depends on whether or not the equipment is used for AAW.

The metadata in Tables 1 and 3 are broken down into Synonym-Homonym Groups (SHG), defined as a collection of two or more attributes that contains at least one pair of synonymous attributes or one pair of homonymous attributes, or both. Attributes not related in any way to synonyms or homonyms are excluded from the group. If more than two attributes are included, each additional attribute must be related to one ormore other attributes of the group. SHGs can be of any size and the number of synonyms and homonyms they can include is not restricted. The concept of the SHG was introduced to focus on both the common ground and the diversity among the component databases and to facilitate the development of a global schema.

Table 1 contains four SHGs, three of which consist entirely of homonyms, whereas Table 3 has 12 SHGs, four of which contain only homonyms, and two of which are composed entirely of synonyms. Out of the 12 SHGs in Table 3, eight groups cover administrative entities. Attributes from six of these groups describe entity identifiers and the other two, CATEGORY and SECURITY, identify aggregates of entities. In contrast, only two groups, LAT and ALTITUDE, pertain to technical characteristics or measurable quantities. In Tables 1 and 3, the SHGs are separated by dotted

Attribute Name	Relation Name	Data Type	Data Length	Attribute Definition
AAW	ALERTTHRES	NUMBER	1	AAW violation threshold in the alert-threshold table
AAW	EIC_NAME_MISSION	CHAR	3	Has AAW mission area?
AAW	THREATAREA	CHAR	10	Level of AAW threat
AMW	ALERTTHRES	NUMBER	1	AMW violation threshold in the alert-threshold table
AMW	EIC_NAME_MISSION	CHAR	3	Has AMW mission area?
AMW	THREATAREA	CHAR	10	Level of AMW threat
ETYPE	EMPSKD	CHAR	1	Employment type in the employment-schedule table
ETYPE	EID	CHAR	8	Equipment type in the equipment-identification table.
HULL	UCHAR	CHAR	6	Hull # of ship or submarine. Squadron # for fixed-wing aircraft and helicopters.
NOSICID	UCHAR	CHAR	5	Identification number assigned to each unit by National Ocean Surveillance Information Center.

Table 1. Synonym-Homonym Groups from the FCCBMP IDB [12]

AAW	anti-air warfare
AMW	amphibious warfare
ASU	anti-surface warfare
ASW	anti-submarine warfare
CCC	command & control
CON	construction
ELW	electronic warfare
FSO	fleet support operations
INT	intelligence
LOG	logistics
MIW	mine warfare
MOB	mobility
NCO	non-combat operations
SPW	special warfare
STW	strike warfare

Table 2. Attributes representing warfare mission areas in the FCCBMP IDB [12]

Attribute Name	Relation Name	Data Type	Data Length	DB	Attribute Definition
ALT	AREA_DEF	REAL	7	OO	Altitude of area
ALTITUDE	PLANS_ELECMBT_TAB	INT	4	NT	Altitude of area
ALTITUDE	TRKPOS	NUMBER	5	OT	Altitude of area
CALLSIGN	TRKID_UNION	CHAR	8	OT	International communications identifier that usually identifies the unit uniquely
CALLSIGN	PLANS_MSN_TAB	CHAR	10	NT	Definition under development
CALL_SIGN_INTL	MERCHANT_SHIPS	CHAR	8	ND	International call sign associated with a specific platform or unit
CATEGORY	TRKID	CHAR	3	OT	Code derived from line identifiers of RAINFORM contact report
CATEGORY	Occurs in 95 tables	CHAR	5	M	Functional classification of a facility by its product or the type of activity in which it is engaged... implements stable facility record identification...
CATEGORY	OSS_FORMS_LIST	CHAR	10	OR	Table categories
DATA_LINE_NUM	CASUALTY_PARTS	NUMBER	2	OR	Sequentially identifies each required repair part.
DATA_LINE_NUM	CASUALTY_STRIP	NUMBER	2	OR	Associates milstrip information with each required repair part.
DESCRIPTION	DD_VALID_VALUES	VARCHAR	255	M	Definition not available at this time
DESCRIPTION	ESS_EVENTS	CHAR	255	OR	Description of the event
DESCRIPTION	ESS_EVENT_TYPES	CHAR	21	OR	Description of the type of event
DESCRIPTION	OSS_FACILITIES	CHAR	80	OR	Description of the facility
DESCRIPTION	OSS_FORMS_LIST	CHAR	50	OR	Form description

Table 3. Synonym-Homonym Groups derived from various C^3I data sets in the Naval Warfare Tactical Database [13]

Attribute Name	Relation Name	Data Type	Data Length	DB	Attribute Definition
FLAG	SORTSM_ORGLOCN	CHAR	1	OR	Organic resource flag to indicate that reporting unit has established subordinate reporting units from its own resources.
FLAG	TRKID	CHAR	2	OT	Code designating country, registry, or political entity to which the platform or unit belongs.
NATIONALITY	UNIT_MASTER_REFERENCE	CHAR	2	OR	Nationality
COUNTRY_CODE	Occurs in 13 tables	CHAR	2	M	Country in which the geographic coordinates are located.
COUNTRY_ACQUIRED	IDBUQL	CHAR	2	M	DOD standard country code of the country from which the equipment was acquired.
FLEET_ID	IDBU	CHAR	1	M	Naval fleet to which a unit is assigned.
FLT	FLEET	CHAR	1	OR	Fleet
HULL	ESS_MESSAGE_D_E	CHAR	6	OR	Hull number
HULL	NON_BLUE_UNITS	CHAR	24	OT	Hull number of ship or submarine, squadron number for air squadrons, unit # for other type units
HULL	TRKID	CHAR	24	OT	Hull number of a ship or submarine, squadron # for fixed-wing aircraft.
HULL_NUMBER	IDBUQL	CHAR	15	M	Hull number of a vessel
PENDANT_NUMBER	IDBUQL	CHAR	10	M	Vessel's pendant (side) number.
PENDANT_NBR	Occurs in 3 tables	CHAR	10	ND	Official identifying number assigned to a specific ship or submarine. It is usually painted on side or hull of the vessel, and may also be indicated by a display of signal pendants. Number given is current or most recent know.

Table 3, continued. Synonym-Homonym Groups derived from various C^3I data sets in the Naval Warfare Tactical Database [13]

Attribute	Relation	Data	Data	DB	Attribute
Name	Name	Type	Length		Definition
NAME_UNIT	ESS_ WORKBOOK	CHAR	30	OR	Name, aname, alias or type/hull of suggested unit to fill this item
NAME_UNIT	UNIT_MASTER_REFERENCE	CHAR	30	OR	Unit designation ... the assigned, abbreviated name of the official designation of an organization.
LAT	PORTS	NUMBER	8	OF	Latitude
LATITUDE	Occurs in 51 tables	CHAR	6	M	Geographic latitude in degrees, minutes and seconds.
SECUR	SORTS_SPCAP	CHAR	2	OR	Security code
SECUR	SORTS_TSKCD	CHAR	2	OR	Security classification
SECURITY	TRKID	CHAR	2	OT	Security classification
TEXT	IDBR_TEXT	CHAR	65	M	Free remarks concerning the entities contained in the IDB UNIT, SITE INSTALLATION, FACILITY, POPULATION AND EQUIPMENT FILES.
TEXT	SORTS_INT_ERRORS	CHAR	255	OR	New description of the error codes
TEXT_RMKS	Occurs in 8 tables	CHAR	60	ND	Remarks pertaining to the description of an entity or item.

Table 3, continued. Synonym-Homonym Groups derived from various C^3I data sets in the Naval Warfare Tactical Database [13]

M	MIIDS/IDB
ND	NID
NT	NTCS-A
OF	OSS IDB - Fixed-site section
OR	OSS IDB - Readiness section
OO	OSS IDB - OPAREA section
OT	OSS IDB - Track section

Table 4. Key to the database (DB) designation in Table 3.

lines. The examples of SHGs from NWTDB shown in Table 3, do not constitute an exhaustive list.

As shown in Table 1, homonyms resulted when tables from databases designed with different purposes in mind, were combined in the FCCBMP IDB [12]. Although anti-air warfare (AAW) and amphibious warfare (AMW) are the only warfare mission area listed here, homonyms were found for attributes signifying 13 other warfare mission areas of the Navy, listed in Table 2. The anti-air warfare attribute, AAW, had a different, although related, meaning, depending on the relation in which it occurred. The same was observed for AMW and all of the other mission warfare areas. Whereas the data element names are the same for the mission warfare area (AAW, AMW, etc.) and ETYPE attributes, the length, nullness, and definitions differ. Because the data type and length combinations are unique for each table, the error checking routine of a DBMS query processor would prevent any attempted joins between these tables on those attributes.

The logical resolution to this homonym problem was to rename the conflicting attributes. Depending on the type of integration required, this solution could be cost prohibitive since it may require a developer to rewrite the application software that access these tables. This is why the ETYPE problem in the FCCBMP IDB could not be resolved. This example supports the observation that the obvious solution from a logical standpoint cannot always be implemented in a practical sense. Sometimes the best that can be done is to inform the users of the semantic differences [6].

Examples of class-one synonyms are the attributes, TRKID.FLAG and COUNTRY_CODE in Table 3. Both attributes refer to an abbreviation designating the country to which a platform, unit, or installation belongs [12]. The data type, length, domain, and length is the same for TRKID.FLAG as for COUNTRY_CODE. Ignoring programming and applications input considerations, the attribute name, "COUNTRY_CODE" could be renamed "FLAG", with no loss of information and no introduction of inconsistency in the results of applications that access these attributes in the database. Similarly, TRKID.FLAG and NATIONALITY are class-one synonyms having the same data type and length. An essential difference between class-one and class-two synonyms is that renaming of a synonym in class one, would not result in the loss of information, whereas the renaming of attributes in class two would cause a loss of metadata, and could affect applications if not modified to accommodate the change. It is for this reason that class-one synonyms are relatively simple to resolve as compared with those in class two.

For example, the class-two synonyms, HULL and NOSICID, which are ship identifiers, both refer to the same specific ship. However, HULL has a data type, length, range, and domain that differ from those of NOSICID. Thus the attribute name, "HULL" could not be renamed "NOSICID" without further modifications, because these two attributes have different formats, and are used for different purposes, even though both identify ships uniquely. The problems with level-two synonyms are similar to those encountered with attributes that have the same definition, but use different units. For example, LENGTH_FT could be an attribute for a ship's length expressed in feet, whereas LENGTH_M could express the same quantity in meters.

Also in reference to Table 3, the PENDANT_NUMUBER and PENDANT_NBR SHG were included in the SHG with HULL to illustrate another class-two synonym pair. HULL and PENDANT_NUMUBER are class-two synonyms with different data domains that point to the same real-world entity, namely vessel. PENDANT_NUMUBER and PENDANT_NBR could be removed from this SHG to form a separate SHG, particularly if the domains of PENDANT_NUMUBER and PENDANT_NBR are very different from those of HULL and HULL_NUMBER, and if there is no one-to-one correspondence. This demonstrates that the manner in which attributes are collected into SHGs is not always clear, and certainly not unique. SHG formation will depend on the kind of analysis being performed.

So far, the discussion in this section has been limited to semantic heterogeneity among attributes, however similar conflicts can occur between relations and attributes. For example, AREA is the name of a relation used to define geographical regions in the OSS IDB whereas AREA is also the name of an attribute in three relations in the OSS IDB.

Below the schema level, domain heterogeneity exists between some tables in OSS and in NTCS-A. This is not always trivial because if the ranges or domains are unequal, the manner in which data elements are integrated will not always be transparent. Even when relations and attributes have the same name, other kinds of heterogeneity such as type heterogeneity, (CHARACTER vs. NUMBER data types) exist for an attribute. Numerous examples of this can be found in Tables 1 and 3.

V. SEMANTIC CONFLICT RESOLUTION ALGORITHMS AND HEURISTICS

Introduction to the Algorithms and Features of the Methodology

In this section, a methodology is described for identifying and resolving semantic heterogeneity using algorithms and heuristics. Each phase of this methodology is based on one of the sublevels of the semantic level heterogeneity shown in Figure 1. Algorithms and heuristics are presented in the form of trouble-shooting flow charts, using a hypothetical example of data dictionary integration between the local schemas of two component databases, A and B. The objective of the algorithms is to construct a global schema for databases A and B to integrate them into a tightly coupled federated database system. These algorithms can be generalized to apply to the schemas of any number of component databases in a federation, and are useful in identifying all of the SHGs present in the aggregate of the component databases. The methodology also can be extended to the case of a fully-merged database by applying the heuristics to the databases as well as to the metadata.

The algorithms captured in the flow charts presented as Figures 2, 3, and 4 were designed to identify and resolve a hierarchy of semantic conflicts, some of which can be resolved at the data dictionary comparison level, and some of which will require an analysis of the data fill and/or specific domain knowledge at schema-definition time. One advantage of these flow charts is that they describe a systematic procedure designed to ensure that the analyst will not omit inadvertently the important steps in the comparisons between relations, attributes, and data fill of the component databases.

As far as possible, the methodology was designed so that semantic inconsistencies would be solved at each higher sublevel before progressing to the next lower sublevel. One should proceed to the next sublevel only when finished at the higher one, or when information is needed from a lower level in order to complete the analysis at the higher sublevel. The flow charts were intended to be applied recursively until each instance of heterogeneity is resolved.

These flow charts are self explanatory, except for the following points: The rectangular boxes represent an action to be performed, including heuristics. Boxes with bold, rounded corners are used to indicate the starting point at each sublevel. Boxes with bold borders signify a logical transition to the next sublevel. The diamonds represent decision points and branches in the procedure. The diamonds with double lines are a reminder that steps need to be performed recursively, until all semantic conflicts have been resolved. The two round-cornered boxes with thin-lined borders in Fig. 3 indicate that the analysis should not or cannot continue at this sublevel, and the procedure for dealing with the situation is outside the scope of this work.

Heuristics

To develop this model for database integration, the methodology included, for example, the following heuristics:

1. Compare the attribute names and scope of definitions before comparing the ranges & domains.

2. If definitions and names differ completely, do not merge (trivial case).

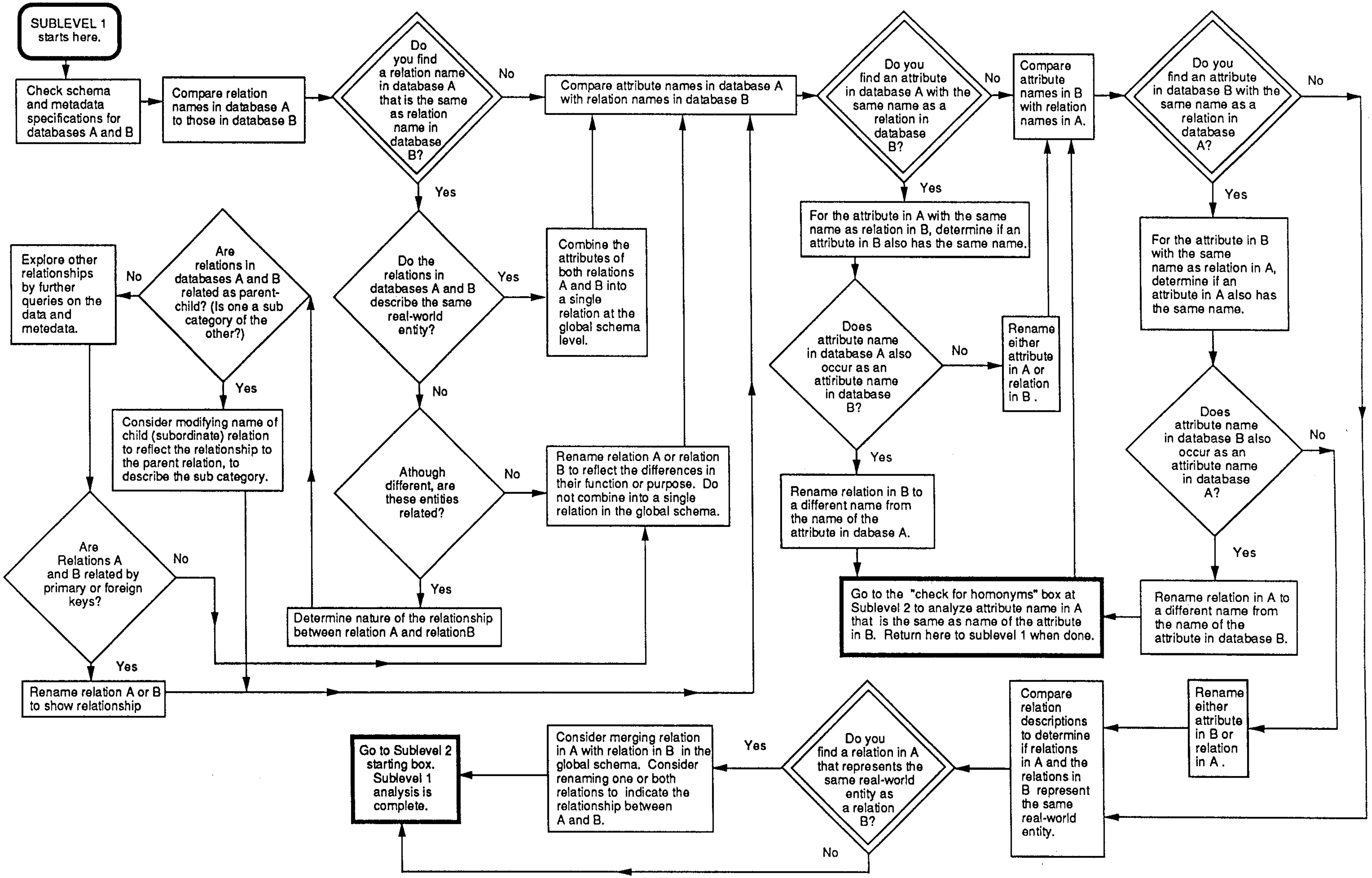
SUBLEVEL 1 starts here.
Check schema and metadata specifications for databases A and B
Compare relation names in database A to those in database B
Do you find a relation name in database A that is the same as relation name in database B?
No
Yes
Compare attribute names in database A with relation names in database B
Do you find an attribute in database A with the same name as a relation in database B?
No
Yes
Compare attribute names in B with relation names in A.
Do you find an attribute in database B with the same name as a relation in database A?
No
Yes
Do the relations in databases A and B describe the same real-world entity?
Yes
No
Combine the attributes of both relations A and B into a single relation at the global schema level.
Athough different, are these entities related?
No
Yes
Rename relation A or relation B to reflect the differences in their function or purpose. Do not combine into a single relation in the global schema.
Determine nature of the relationship between relation A and relationB
Are relations in databases A and B related as parent-child? (Is one a sub category of the other?)
No
Yes
Explore other relationships by further queries on the data and metedata.
Consider modifying name of child (subordinate) relation to reflect the relationship to the parent relation, to describe the sub category.
Are Relations A and B related by primary or foreign keys?
No
Yes
Rename relation A or B to show relationship
For the attribute in A with the same name as relation in B, determine if an attribute in B also has the same name.
Does attribute name in database A also occur as an attiribute name in database B?
No
Yes
Rename either attribute in A or relation in B .
Rename relation in B to a different name from the name of the attribute in dabase A.
Go to the "check for homonyms" box at Sublevel 2 to analyze attribute name in A that is the same as name of the attribute in B. Return here to sublevel 1 when done.
For the attribute in B with the same name as relation in A, determine if an attribute in A also has the same name.
Does attribute name in database B also occur as an attiribute name in database A?
No
Yes
Rename relation in A to a different name from the name of the attribute in database B.
Rename either attribute in B or relation in A .
Compare relation descriptions to determine if relations in A and the relations in B represent the same real-world entity.
Do you find a relation in A that represents the same real-world entity as a relation B?
Yes
No
Consider merging relation in A with relation in B in the global schema. Consider renaming one or both relations to indicate the relationship between A and B.
Go to Sublevel 2 starting box. Sublevel 1 analysis is complete.

Figure 2. Trouble-shooting flow chart for Sublevel 1, Relations

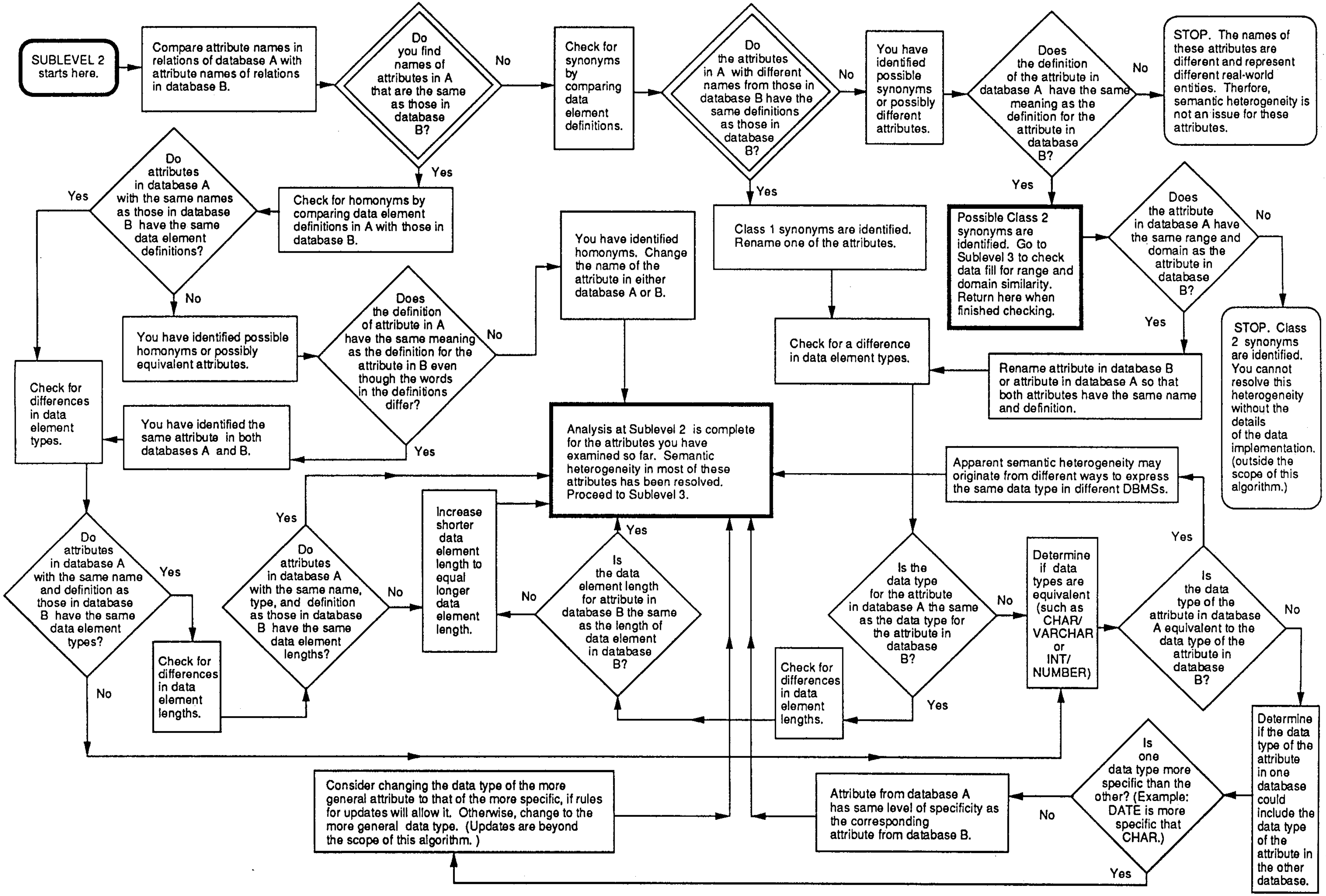
SUBLEVEL 2 starts here.
Compare attribute names in relations of database A with attribute names of relations in database B.
Do you find names of attributes in A that are the same as those in database B?
No
Yes
Check for synonyms by comparing data element definitions.
Do the attributes in A with different names from those in database B have the same definitions as those in database B?
No
Yes
You have identified possible synonyms or possibly different attributes.
Does the definition of the attribute in database A have the same meaning as the definition for the attribute in database B?
No
Yes
STOP. The names of these attributes are different and represent different real-world entities. Therfore, semantic heterogeneity is not an issue for these attributes.
Check for homonyms by comparing data element definitions in A with those in database B.
Do attributes in database A with the same names as those in database B have the same data element definitions?
Yes
No
You have identified possible homonyms or possibly equivalent attributes.
Does the definition of attribute in A have the same meaning as the definition for the attribute in B even though the words in the definitions differ?
No
Yes
You have identified homonyms. Change the name of the attribute in either database A or B.
You have identified the same attribute in both databases A and B.
Class 1 synonyms are identified. Rename one of the attributes.
Possible Class 2 synonyms are identified. Go to Sublevel 3 to check data fill for range and domain similarity. Return here when finished checking.
Does the attribute in database A have the same range and domain as the attribute in database B?
No
Yes
STOP. Class 2 synonyms are identified. You cannot resolve this heterogeneity without the details of the data implementation. (outside the scope of this algorithm.)
Rename attribute in database B or attribute in database A so that both attributes have the same name and definition.
Check for a difference in data element types.
Check for differences in data element types.
Analysis at Sublevel 2 is complete for the attributes you have examined so far. Semantic heterogeneity in most of these attributes has been resolved. Proceed to Sublevel 3.
Apparent semantic heterogeneity may originate from different ways to express the same data type in different DBMSs.
Do attributes in database A with the same name and definition as those in database B have the same data element types?
Yes
No
Check for differences in data element lengths.
Do attributes in database A with the same name, type, and definition as those in database B have the same data element lengths?
Yes
No
Increase shorter data element length to equal longer data element length.
Is the data element length for attribute in database B the same as the length of data element in database B?
Yes
No
Check for differences in data element lengths.
Is the data type for the attribute in database A the same as the data type for the attribute in database B?
No
Yes
Determine if data types are equivalent (such as CHAR/ VARCHAR or INT/ NUMBER)
Is the data type of the attribute in database A equivalent to the data type of the attribute in database B?
Yes
No
Determine if the data type of the attribute in one database could include the data type of the attribute in the other database.
Is one data type more specific than the other? (Example: DATE is more specific that CHAR.)
No
Yes
Attribute from database A has same level of specificity as the corresponding attribute from database B.
Consider changing the data type of the more general attribute to that of the more specific, if rules for updates will allow it. Otherwise, change to the more general data type. (Updates are beyond the scope of this algorithm.)

Figure 3. Trouble-shooting flow chart for Sublevel 2, Attributes

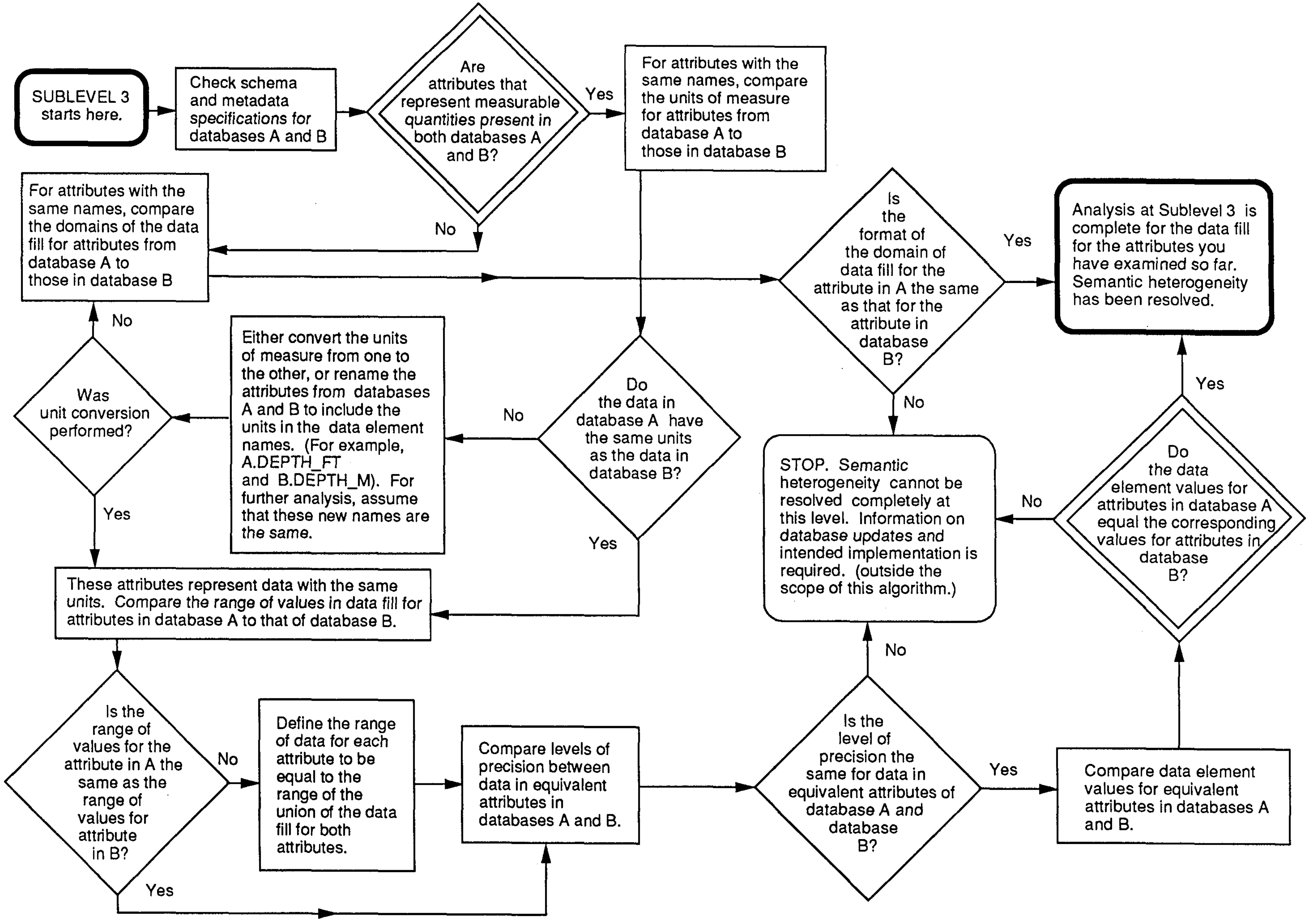
SUBLEVEL 3 starts here.
Check schema and metadata specifications for databases A and B
Are attributes that represent measurable quantities present in both databases A and B?
Yes
No
For attributes with the same names, compare the units of measure for attributes from database A to those in database B
For attributes with the same names, compare the domains of the data fill for attributes from database A to those in database B
Is the format of the domain of data fill for the attribute in A the same as that for the attribute in database B?
Yes
No
Analysis at Sublevel 3 is complete for the data fill for the attributes you have examined so far. Semantic heterogeneity has been resolved.
No
Was unit conversion performed?
Yes
Either convert the units of measure from one to the other, or rename the attributes from databases A and B to include the units in the data element names. (For example, A.DEPTH_FT and B.DEPTH_M). For further analysis, assume that these new names are the same.
No
Do the data in database A have the same units as the data in database B?
Yes
STOP. Semantic heterogeneity cannot be resolved completely at this level. Information on database updates and intended implementation is required. (outside the scope of this algorithm.)
Yes
No
Do the data element values for attributes in database A equal the corresponding values for attributes in database B?
These attributes represent data with the same units. Compare the range of values in data fill for attributes in database A to that of database B.
No
Is the range of values for the attribute in A the same as the range of values for attribute in B?
No
Define the range of data for each attribute to be equal to the range of the union of the data fill for both attributes.
Compare levels of precision between data in equivalent attributes in databases A and B.
Is the level of precision the same for data in equivalent attributes of database A and database B?
Yes
Compare data element values for equivalent attributes in databases A and B.
Yes

Figure 4. Trouble-shooting flow chart for Sublevel 3, Data fill

3. If the definitions differ but the attribute names are the same (homonyms), rename one of the attributes.

4. If definitions are identical but the names differ (synonyms), compare data element type and length, range and domain.

Limitations of the Methodology

This methodology applies only to the semantic level, depicted in Fig. 1, and was not designed for application at the platform or data model levels, although references to these levels in the heuristics are included to delineate the boundary of the algorithms' applicability. These boundaries are not always distinct because of the complexity and ambiguity in the problem to be solved, particularly at Sublevel 3, where the heuristics can be less general and obvious. The methodology includes some decisions on how to deal with semantic heterogeneity that are somewhat arbitrary. This is as it should be, owing to the arbitrary nature in which many attribute and relation names and definitions generally are selected in autonomous databases.

The methodology is predicated upon the assumption that an analyst can make same/difference judgments. Sometimes this is ambiguous, particularly when dealing with class-two synonyms, which could not be resolved at Sublevel 3. Analysis at Sublevel 3 is the most difficult because it is the sublevel closest to the point where knowledge of data updates and implementations is required. Moreover, the resolution of some data-type heterogeneity will depend on update and implementation. For example, if an application requires a specific numerical data type for a given attribute, a format error could result from an update to the attribute if the allowed data type has been relaxed to the more general character data type.

Although this methodology covers several properties of relations, attributes and their data fill, heterogeneity with respect to nullness is ignored. Differences in levels of security and data granularity (as in fleet, ship, squadron) except at sublevel one, were ignored. Moreover, it was assumed that no updates or modifications of any aspect of the component databases would be allowed during the data dictionary analysis and algorithm implementation.

This paper is intended to establish a framework for the systematic resolution of semantic inconsistencies. It is expected that the algorithms captured in Figures 2 through 4 can be refined through usage, and that improvements can be made as a result of experience gained in actual database integration situations. Because of the variety and complexity of semantic problems, like other proposed solutions, this methodology is appropriate for resolving some, but not all semantic inconsistencies.

Lastly, an implicit assumption behind this methodology is that the controlling authorities of component databases are willing to cooperate. No technical solution concerning database and data-dictionary integration will be useful if political forces preclude its implementation.

VI. CONCLUSIONS AND DIRECTIONS FOR FUTURE RESEARCH

A systematic approach to the resolution of semantic heterogeneity has been developed and illustrated with examples derived from the component C^3I system databases in a federation. This approach was designed to develop a global schema for integrating two C^3I system databases; however, it is general enough to apply to a variety of other database integration situations.

More work is needed in this area. A flow chart at an additional sublevel can be constructed to address conflicts arising from data updates and intended use. Although these algorithms can identify class-two synonyms, a better way to resolve them is needed.

As indicated above, the algorithms presented here are preliminary and need to be modified and refined as more experience is gained through the usage in actual database integration efforts. Ultimately, these algorithms need to be incorporated in automated tools to aid database integrators in their integration efforts. Because semantics originate in the minds of the various database developers, the resolution of all problems with semantic heterogeneity in database integration cannot be fully automated; an analyst will be required to evaluate some data conflicts and formulate solutions based on familiarity with the semantics of the application domain and implementation.

VII. REFERENCES

1. Sheth, Amit P. and J. A. Larson, "Federated Database Systems for Managing Distributed, Heterogeneous and Autonomous Databases", *ACM Computing Surveys,* 22(3), 183 - 236, September, 1990.

2. Ventrone, Vincent and Sandra Heiler, "Semantic Heterogeneity as a Result of Domain Evolution", *SIGMOD Record* 20(4), 16 - 20, December, 1991.

3. Drew, Pam, Roger King, Dennis McLeod, Marek Rusinkiewicz and Avi Silberschatz, "Report of the Workshop on Semantic Heterogeneity and Interoperation in Multidatabase Systems", *SIGMOD Record* 22(3), 47 - 56, September, 1993.

4. Breitbart, Y., "Multidatabase Interoperability", *SIGMOD Record* 19(3), 53 - 60, September, 1990.

5. Sheth, Amit P., "Semantic Issues Multidatabase Systems", *SIGMOD Record* 20(4), 5 - 9, December, 1991.

6. Kent, William, "The Breakdown of the Information Model in Multi-database Systems ", *SIGMOD Record* 20(4), 10 - 15, December, 1991.

7. Kamel, Magdi N., "Identifying and Resolving Semantic Conflicts in Distributed Heterogeneous Databases", *Proceedings of the Tenth Annual DOD Database Colloquium '93*, San Diego, 25 Aug. 1993.

8. Thuraisingham, Dr. Bhavani M., *Security Issues for Federated Database Systems to Manage Distributed, Heterogeneous and Autonomous Multilevel Databases,* Mitre Corp. M91-78,10 - 13, November, 1991.

9. Brathwaite, Dr. Kenmore S., *Relational Databases - Concepts Design and Administration* , McGraw-Hill, New York, NY., 6-7, 1991.

10. Ceruti, Dr. Marion G., Sharon D. Rotter, Kristofer Timmerman and Jennifer Ross, "Operations Support System (OSS) Integrated Database (IDB) Design and Development: Software Reuse Lessons Learned", *Proceedings of the AFCEA Database Colloquium '92*, 25 - 27 Aug. 1992.

11. Litwin,W., J. Boudenant, C. Esculier, A. Ferrier, A. Glorieux, J. La Chimia, K. Kabbaj, C. Moulinoux, P. Rolin, and C. Stangret, "SIRUS Systems for Distributed Data Management" in *Distributed Data Bases*, H. -J. Schneider, Ed., North-Holland, The Netherlands, pp. 311-366, 1982.

12. Ceruti, M. G. and J. P. Schill, "*Fleet Command Center Battle Management Program (FCCBMP) Data Dictionary*", Version 3, NOSC TD 1320, September, 1988.

13. Batini, C., S. Certi, and S. B. Navathe, *Conceptual Database Design: An Entity-Relationship Approach*, Benjamin/Cummings Publishing Co., 1992.

14. Department of the Navy, Office of the Chief of Naval Operations, *Naval Warfare Tactical Database (NWTDB) Standards Manual, Vol. 2: Data Element Dictionary, Version 2, (draft)*, March, 1994.

Dr. Marion G. Ceruti

NCCOSC RDT&E Division, Code 4222
53140 Gatchell Rd., San Diego, CA 92152-7464
Tel. (619) 553-4068, DSN 553-4068, Fax (619) 553-5136
MILNET: ceruti@nosc.mil

Dr. Ceruti is a scientist in the Systems Integration Group of the Command and Intelligence Systems Division at the Naval Command, Control and Ocean Surveillance Center, Research, Development, Test, and Evaluation Division. She received her Ph.D. in Physical Chemistry, with emphasis on data acquisition systems, from the University of California at Los Angeles (UCLA) in 1979. While at UCLA, she was awarded a research fellowship from the International Business Machine Corp. Her present professional activities include database development and integration for C^4I decision support systems, including the Operations Support System and the Joint Maritime Command Information System. Dr. Ceruti has served on the program committee and as Government Point of Contact for all annual Database Colloquia since 1987. An active member of AFCEA and several other scientific and professional organizations, she is the author of numerous publications on various topics in science and engineering, including information management.

Dr. Magdi N. Kamel

Department of Systems Management
Naval Postgraduate School
555 Dyer Rd., Monterey, CA 93943
Tel. (408) 656-2494, Fax (408) 656-3407
INTERNET: kamel@nps.navy.mil

Dr. Kamel is an Associate Professor in the Information Systems Group in the Naval Postgraduate School. He received his Ph.D. in Information Systems from the Wharton School, University of Pennsylvania. His main research interests include database management systems, specifically data models and languages, interoperability, and integration issues in heterogeneous databases, and the integration of databases with expert and decision support systems. Dr. Kamel is frequently invited to present papers on this subject at meetings and conferences. He has consulted in these areas for several organizations and is the author of numerous published research papers on database management topics. Dr. Kamel is a member of Association for Computing Machinery and the IEEE Computer Society.

21a. NAME OF RESPONSIBLE INDIVIDUAL M. G. Ceruti	21b. TELEPHONE *(include Area Code)* (619) 553–4068	21c. OFFICE SYMBOL Code 4222